COMPUSCAN 170

CODE CONVERSION HEADER SHEET MANUAL

6/13/73

REV HP3 & UP

>8K

TABLE OF CONTENTS

		Page
I.	GENERAL DESCRIPTION	1
II.	HEADER SHEET FORMAT	2
III.	OPERATING INSTRUCTIONS	4
IV.	ERROR MESSAGES	7
V.	SPECIAL SUBROUTINES	8
	APPENDIX A - HEADER SHEETS	I
	APPENDIX B - DESCRIPTION OF CONVERSION TABLES	II

I. GENERAL DESCRIPTION

The scanning program, supplied by CompuScan with the Model 170 Optical Page Reader, contains a set of code conversions. Further conversions can be defined or an entirely new set of conversion rules can be entered conveniently using the Code Conversion Header Sheets.

The Header Sheets are typed in a simple, well readable format. For instance, to enter the conversion rule, that the characters /v should be converted to unshift 44 ("add thin") is indicated by the following equation on the Header Sheet:

/v = U44

The Header Sheets have to be scanned using the Header Sheet Routine. In the process of this, the routine decodes the equations typed on the sheets and stores the appropriate conversion rules in the memory of the Nova computer, in the form of the so called "conversion tables".

The last conversion definition typed on the Header Sheet is followed by a ΔΔ, which terminates the Header Sheet operation.* Now the scanner is ready for scanning ordinary text, unless Paragraph Insertion or Typescan is required. In the latter case, these routines have to be loaded, since the Header Sheet Routine occupies the same area in the memory of the Nova.

Optionally, the user may store the new conversion rules on paper tape or magnetic tape. This is desireable, if the new conversions are regularly used.

---

*The ΔΔ must be typed on a new line.

II. HEADER SHEET FORMAT

1) Each code conversion is specified by a statement of the form:

⟨INPUT⟩ ⟨1 or more spaces⟩ = ⟨Ø or more spaces⟩ ⟨OUTPUT⟩

⟨INPUT⟩ is any string of scanable characters to be converted to an output code on paper tape or magnetic tape. No embedded spaces are permited in the INPUT. Instead, the notation ΔnS has to be used to represent n spaces, where n is a decimal number (less than 16). The notation ΔS represents one space.

If a character (e.g. a preprinted symbol) cannot be typed, it may be indicated in the input string by 'ΔnA', where n is one, two or three octal digits to represent the ASCII code of the character (see 'CHARACTER SET' section in the Scanning Program Summary).

The special context 'At the end of the line' is indicated by the form ΔE, the special context 'At the beginning of the line' by the form ΔB.

OUTPUT is a sequence of one or more output specifications to which the corresponding input string is converted. The following output specifications are defined:

Sdd specifies a SHIFT mode TTS code (dd is a two-digit octal number).

Udd specifies an UNSHIFT mode TTS code (dd is a two-digit octal number).

Xhh specifies an 8-bit output code (hh is attwo-digit hexadecimal number).

I specifies that the input is to be ignored.

*dd specifies that special subroutine number dd is to be invoked (dd is a two-digit octal number).

Output specifications are separated by commas with no embedded spaces permitted.

Example 1: The Header Sheet Statement

```
△2S'' = U04,S42,S42
```

defines the conversion of two spaces followed by two quotation marks into a space band and two open quotes.

Example 2: The Header Sheet Statement

```
{△E = I
```

defines that the character { at the end of a line must be ignored.

Example 3: The Header Sheet Statement

```
△B△5S/ = *12
```

specifies that the special subroutine #12 must be called, when five spaces at the beginning of line followed by a quote are scanned

Example 4 : The Statement

```
A = S30
```

specifies that the letter A be output as "Shift 30" TTS code.

Example 5 : The Statement

```
A = XC1
```

defines that the standard EBCDIC code (hex C1 or octal 301) be output for the letter A.

Example 6 : The Statement

```
A = A101
```

defines that the standard ASCII code (octal 101) be output for the letter A.

2) Lines beginning with △ followed by a space are comments and do not effect the stored conversion rules in any way.

3) The line △△ causes the Header Sheet routine to "close" the conversion tables built up in the Nova memory, and terminates the Header Sheet mode.

4) A conversion line that ends with a comma will be continued on the next line.

III. OPERATING INSTRUCTIONS

A) To build up a new conversion table from scratch, the following steps are necessary:

1. Load the standard scanning program and reference alphabet.
2. Load the Header Sheet Routine using the binary loader.
3. Initialize the Header Sheet Routine by starting the program at 13000. The message CODES CLEARED will appear on the CRT and the HEADER MODE light will be turned on. Header mode is automatically entered.
4. Start scanning by starting the program at 400.
5. Use the appropriate Operator Commands to set parameters required for scanning the Header Sheets: disable Scan Edit, enable Key Optics, set required line spacing (lines/inch), and enable punch output (loc. 45 = 1).
6. Manually store 0 or 1 in location 47 (1=6-bit TTS code, 0=8-bit output code).
7. Proceed to scan the header sheets. The HEADER MODE light should go on after the first line is scanned.
8. Header Mode will be terminated where a ΔΔ line is encountered or a manual transfer to 13004 is executed.
9. When Header Mode is terminated, normal scanning using the new conversion tables may begin immediately, unless Paragraph Insertion is required. (See Section C).

B) To modify a code conversion table in the core memory of the Nova, the following steps are necessary:

1. Load the standard scanning program and reference alphabet.
2. Load the Header Sheet Routine using the binary loader.
3. Start the program at 13010. The reading head moves to the top margin position and the FEED light goes on.

4. Use the appropriate Operator Commands to set parameters required for scanning the Header Sheets: disable Scan Edit, enable Key Optics (recommended, but not absolutely necessary), set line spacing, and enable punch output (loc. 45 = 1).

5. Manually store 0 or 1 in location 47. 6-bit TTS code: 1; 8-bit output code: 0.

6. Proceed to scan header sheets with code conversions. The HEADER MODE light should go on after the first line is scanned.

7. Header Mode will be terminated, when a ΔΔ line is encountered or the program manually started at 13004.

8. Normal scanning can be started immediately, unless Paragraph Insertion is required (See Section C).

C) If Paragraph Insertion is required, the Paragraph Insertion Routine has to be loaded after the completion of the Header Sheet operation. This is necessary, since the Header Sheet Routine and the Paragraph Insertion Routine occupy the same area in the core memory of the Nova computer. Paragraph Insertion (PINS) is supplied on separate tape and loaded using the binary loader. PINS is also included in the standard scanning program tape; therefore, it is not necessary to load PINS separately after loading the scanning program.

D) If the new conversions are to be used regularly, it is desirable to save the conversion tables on paper tape or magnetic tape. To accomplish this, the following locations of the Nova core memory have to be written on tape:

1. Location 47
2. Locations 302 thru 304, and 324 thru 325.
3. Locations C(302) thru C(324) where C(302) and C (324) are the contents of locations 302 and 324, respectively.

The program DEBUG II, or the program PABST, can be used to punch the above locations on 8-level paper tape, and the program WABST to write on magnetic tape.

Example: Let us assume that after scanning the header sheets, location 302 contains 20000 and location 324 contains 22000. The following DEBUG II commands are required to save the conversion tables on 8-level paper tape:

```
47,47 P
302,304 P
20000,22000 P
324,325 P
E
```

The "E" command is required to punch an "end block", which is required at the end of any binary tape.

DEBUG II is supplied on a separate paper tape and is loaded using the binary loader. Since it partially overlays the scanning program, the latter has to be reloaded after punching the code conversion table tape.

PABST is supplied on a separate paper tape and is loaded using the binary loader. It loads into a data buffer and does not affect the scanning buffer; however, execution of the scanning program destroys it.

IV. ERROR MESSAGES

If an error is detected during the scanning of a header sheet, the program displays and error message on the CRT:

CODES ERROR: X

where X is a decimal digit, or the character 'M' or 'N'.

The explanation of the error message numbers is given in the following table.

Displayed Error No.	Action By Program	Cause
1	I	Illegal input string : nonspecific cause, possibly recognition error.
2	I	Illegal blanks : blanks appear in contexts other than adjacent to = sign.
3	I	Illegal S or U specifications
4	I	Illegal X specifications
5	I	Illegal * specifications
6	I	Illegal output string specification
7	H	NSPCE table full, no more room in dynamic core.
8	H	MULTIN table full, no more room in dynamic core.
0	H	Dynamic core full, tables can't be expanded.

M -- MULTN codes exceed maximum allowable size of table, processing stops, line not entered.

N -- NSPCE codes exceed maximum allowable size of table, processing stops, conversion not completed.

ACTIONS : I - Line is ignored, processing continues.

H - Line was entered, no more space available, processing stops.

V. SPECIAL SUBROUTINES

There are 9 output "subroutines" in the standard Model 170 scanning program, that can be invoked by the header sheet expression "*dd" where dd is the subroutine number.

The "typical input" items shown in the below table are examples only. The input strings to call certain special subroutines are defined through header sheet statements.

The "output functions" shown in the table, however, cannot be changed using the present version of the Header Sheet Routine. Furthermore, the output functions implemented in a particular scanning program tape may differ from those shown here to fit requirements of the installation. The exact output codes produced by the special subroutines in a given scanning program can be determined from the Scanning Program Summary, which is individually prepared for each scanning program tape.

Subroutine Number	Interpretation	Typical Input	Output Function
00	End Take	$t(CR)	U66,U72 + 10" Leader
01	Indent Logic	5 spaces	U66,U64 if Special End Flag is reset. U64 if special end flag is set The Special End Flag is unconditionally reset.
04	Quad Left	/l (CR)	U66 Special End Flag is set.
05	Quad Center	/c (CR)	U57 Special End Flag is set.
06	Quad Center	/r (CR)	U76 Special End Flag is set.
10	End of Line	(CR)	Reset Special End Flag.
11	Indented Quotes (Open Quotes)	ØØØØØ''	U66,U64,S42,S42 if Special End Flag is Reset. U64,S42,S42 is Special End Flag is set.
12	Indented Quote (Open Quote)	ØØØØØ'	U66,U64,S42 if Special End Flag is Reset. U64 S42 if Special End Flag is set.
17	Punch Leader	/T	10" of Leader

APPENDIX A

Sample Header Sheets

JOB SAMPLE HEADER SHEET NO PAGE

ALIGN FIRST CHARACTER UNDER THIS ARROW

```
A HOT METAL CHARACTERS ALPHABETIC, UPPER CASE 7-26-72
A = S30
B = S23
C = S16
D = S22
E = S20
F = S26
G = S13
H = S05
I = S14
J = S32
K = S36
L = S11
M = S07
N = S06
O = S03
P = S15
Q = S35
R = S12
S = S24
T = S01
U = S34
V = S17
W = S31
X = S27
Y = S25
Z = S21
```

DO NOT TYPE BELOW THIS LINE

ALIGN FIRST CHARACTER UNDER THIS ARROW

```
Δ HOT METAL CHARACTERS, ALPHABETIC, LOWER CASE 7-26-72
a = U30
b = U23
c = U16
d = U22
e = U20
f = U26
g = U13
h = U05
i = U14
j = U32
k = U36
l = U11
m = U07
n = U06
o = U03
p = U15
q = U35
r = U12
s = U24
t = U01
u = U34
v = U17
w = U71 ¤¤¤w ¤¤¤w = U31
x = U27
y = U25
z   U21 ¤¤¤z = U21
```

DO NOT TYPE BELOW THIS LINE

JOB SAMPLE HEADER SHEET NO PAGE

ALIGN FIRST CHARACTER UNDER THIS ARROW

△ HOT METAL CHARACTERS, NUMERICS & SPECIALS 7-26-72

0 = U55
1 = U67
2 = U71
3 = U60
4 = U52
5 = U41
6 = U65
7 = U74
8 = U54
9 = U43
! = S70
$ = U70
& = S43
(= S61
) = U61
+ = U04 SPACE BAND
, = U46
? = S55
– = U62 HYPHEN
. = U47
: = S53
; = U53

DO NOT TYPE BELOW THIS LINE

ALIGN FIRST CHARACTER UNDER THIS ARROW

```
1  ▵ HOT METAL CHARACTERS: QUOTES,INDENTED QUOTES
   ▵ & BLANK COMPRESSION.  7-26-72
   ▵¤
2  ▵ BLANK COMPRESSION & OPEN QUOTES
   ▵4S/ = U04,S42
3  ▵3S/ = U04,S42
   ▵2S/ = U04,S42
   ▵1S/ = U04,S42
4  ▵4S// = U04,S42,S42
   ▵3S// = U04,S42,S42
5  ▵2S// = U04,S42,S42
   ▵1S// = U04,S42,S42
6  ▵¤
   ▵ QUOTES AT BEGINNING OF LINE(OPEN)
   ▵B/ = S42
7  ▵B// = S42,S42
   ▵¤
8  ▵ FORCE OPEN QUOTES
   =/ = S42
   =// = S42,S42
9  ▵¤
   ▵ CLOSE QUOTES
10 / = U42
   // = U42,U42
11 ▵¤
   ▵ CLOSE QUOTES
   / = U42
12 // = U42,U42
   ▵¤
13 ▵ FORCE CLOSE QUOTES
   $/ = U42
14 $// = U42,U42
   ▵¤
   ▵ INDENTED QUOTES
15 ▵B▵6S/ = *12
   ▵B▵5S/ = *12
16 ▵ WHERE *12 PUTS SAME AS *12¤ + S42
   ▵B▵6S// = *11
17 ▵B▵5S// = *11
   ▵ WHERE *11 PUTS SAME AS *1 + S42,S42
   ▵¤
18 ▵ BLANK COMPRESSION
   ▵4S = U04
19 ▵3S = U04
   ▵2S = U04
   ▵1S = U04
20
21
22
23
```

DO NOT TYPE BELOW THIS LINE

JOB SAMPLE HEADER SHEET NO PAGE

ALIGN FIRST CHARACTER UNDER THIS ARROW

```
1  △ HOT METAL CHARACTERS, MISCELLANEOUS  7-26-72
   △¤¤¤
2  △ SLASH STRINGS
   /// = U64              EM SPACE
   // = U56               EN SPACE
3  /+ = U40               THIN SPACE
   /.. = U45              EM LEADER
4  /. = U75               EN LEADER
   /R = U77               RUB-OUT
   /t = U00               TAPE FEED
5  /Ss = S33              FORCE SHIFT
   /Uu = U37              FORCE UNSHIFT
6  △ FRACTIONS
   /18 = S67              1/8
7  /12 = S52              1/2
   /34 = S65              3/4
   /14 = S71              1/4
8  /38 = S60              3/8
   /58 = S41              5/8
9  /78 = S74              7/8
   △¤¤¤
   △ END OF NORMAL LINE
10 △E = *10
   △ WHERE *10 SETS SP. END FLAG=0
11 △¤¤¤
   △ IGNORE SPACE AT END OF LINE
12 {△E = I
   △ TWO HYPHENS (= EM DASH)
   -- = S54
13 △¤¤¤
   △ UPPER & LOWER RAIL
14 { = U63                UPPER RAIL
   } = U73                LOWER RAIL
15
16
17
18
19
20
21
22
23
```

DO NOT TYPE BELOW THIS LINE

JOB SAMPLE HEADER SHEET NO PAGE

ALIGN FIRST CHARACTER UNDER THIS ARROW

```
1  △ HOT METAL COMMANDS  8-1-72
   △¤¤¤
2  △ AUTO INDENTS
   △B△6S = *1
3  △B△5S = *1
   △ WHERE *1 OUTPUTS U66,U64 IF SP. END FLAG=0, ELSE U64 ONLY
4  △¤¤¤
   △ PAGE TERMINATION
5  $t△E = *0            END OF TAKE
   △ WHERE *0 PUTS U66(QL),U72(BELL),PAGE TERMINATE
6  △ & 10// TAPE LEADER
   %%△E = *2
7  △ WHERE *2 TERMINATES PAGE & PUTS 10// TAPE LEADER
   △¤¤¤
8  △ PUNCH LEADER
   /T = *17             PUNCH 20¤¤10// OF TAPE FEED
9  △ WHERE *17 OUTPUTS 10// TAPE FEED ONLY
   △ QUADS
10 /l = U66             QUAD LEFT
   /l△E = *4
11 △ WHERE *4 PUTS U66 AND SETS SP. END FLAG=1
   /c = U57             QUAD CENTER
12 /c△E = *5
   △x¤ WHERE *5 PUTS U57 AND SETS SP. END FLAG=1
13 /r = U76             QUAD RIGHT
   /r△E = *6
14 △ WHERE *6 PUTS U76 AND SETS SP. END FLAG=1
   △¤¤¤
15 △ SLASH STRINGS
   /v = U44             ADD THIN
16 /v△E = U44
   /e = U10             ELEVATE
17 /e△E = U10
   /m = U51             VERTICAL RULE
18 /m△E = U51
   /p = U50             PAPER FEED
19 /p△E = U50
   /b = U72             BELL
20 /b△E = U72
   △¤¤¤
21 △ EQUA:S STRINGS¤¤¤
   △ EQUALS STRINGS
22 = = U02              RETURN
   =△E = U02
23
```

DO NOT TYPE BELOW THIS LINE

[illegible] CONTACTS UNDER THE ARROW

```
*12* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U67,U54,
U55,U10,U26,U67,U71,U55,U10,U13,U71,U66,U73      12 PT.HEAD
*12i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U67,U54,
U55,U10,U26,U67,U71,U55,U10,U13,U71,U66,U63      12 PT.ITALIC HEAD
*14* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U71,U55,U55,~~U10,U26,~~
U10,U26,U67,U52,U55,U10,U13,U71,U66,U73      14 PT.HEAD
*14i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U71,U55,
U55,U10,U26,U67,U52,U55,U10,U13,U71,U66,U63      14 PT.ITALIC HEAD
*18* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U71,U52,U55,~~U10~~
U10,U26,U67,U54,U55,U10,U13,U71,U66,U73      18 PT.HEAD
*18i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U71,U52,
U55,U10,U26,U67,U54,U55,U10,U13,U71,U66,U63      18 PT.ITALIC HEAD
*23¤4* = U10,U43,U43m¤,U10,U30,U52,U71,U55,U10,U23,U60,U55,
U55,U10,U26,U71,U52,U55,U10,U13,U71,U66,U73      24 PT.HEAD
*24i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U60,U55,
U55,U10,U26,U71,U52,U55,U10,U13,U71,U66,U63      24 PT.ITALIC HEAD
*30* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U60,U65,U55,~~U10,U26,~~
U10,U26,U60,U55,U55,U10,U13,U71,U66,U73      30 PT.HEAD
*30i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U60,U65,
U55,U10,U26,U60,U55,U55,U10,U13,U71,U66,U63      30 PT.ITALIC HEAD
*36* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U52,U71,U55,
U10,U26,U60,U65,U55,U10,U13,U71,U66,U73      36 PT. HEAD
*36i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U52,U71,U55,
U10,U26,U60,U65,U55,U10,U13,U71,U66,U63      36 PT.ITALIC HEAD
~~*48* = U10,U43,U43,U10,U30,U52,U71,U66,U73      48 PT.HEAD~~
*48i* = U10,U43,U43,U10,U30,U52,U71,U55,U10,U23,U41,U52,
U55,U10,U26,U52,U54,U55,U10,U13,U71,U66,U63      48 PT.ITALIC HEAD
```

DO NOT TYPE BELOW THIS LINE

APPENDIX B

Description of Conversion Tables

The scanned input characters are internally represented by 7-bit ASCII codes in the scanning program. In the final state of the program, these ASCII "internal characters" are converted to desired output codes.

Three tables define the output codes associated with the internal characters; Character Table (CODES), Multiple Conversion Table (MULTIN) and Explosion Table (NSPCE). The MULTIN and NSPCE tables are expansions of the CODES table.

The CODES table has 128_{10} (200_8) single word entries correxponding to all possible 7-bit octal codes 0-177. The internal character masked to 7 bits is used as an index into the CODES table. Each entry in the table determines the conversion to be applied. Bits 0 and 1 and the magnitude of the entry define the conversion:

Bits 0 - 1	
00	One to one conversion if word ≤ 377.
00	Multiple conversion in second MULTIN table if word >377.
01	One to N conversion (Explosion)
10	Multiple conversion in first MULTIN table.
11	Bits 2 and 3 distinguish between 00 - Special Process Code 10 - Ignore Internal Character 11 - Illegal Internal Character

The one-to-one conversion entry signifies 8 bits of output information in bits 8-15 of the CODES entry. These are either an 8-bit output character or a shift bit (bit 8) and a 6-bit output character as determined by a code flag called SHIFT (loc. 47).

The one-to-N conversion entry signifies a multiple output of characters for the one internal character. These characters are contained in the Explosion Table (NSPCE). The CODES table entry bits 2-6 specify the number of characters to output and bits 7-15 specify the starting character number in table NSPCE (first character to output). Alternately, the CODES table entry may be a special pointer to a 2-word entry in the MULTIN table, which selects an area of the NSPCE table and then outputs the desired character string. The 8-bit characters in the NSPCE table are treated the same as the one-to-one conversion characters.

The multiple conversion entry signifies that the internal character is the first character of several strings of characters designated as single entities for conversion. The strings are contained in the Multiple Conversion Table (MULTIN). The CODES table entry bits 2-6 specify the number of characters and bits 7-15 specify the starting core address relative to the first address of the MULTIN table of the first MULTIN string beginning with the internal character.

The "special process" code entry signifies a special subroutine is to be called. The "ignore internal character" entry signifies an internal character presented for output that has not been designated as a legal character for the set of internal codes.

The MULTIN table has variable length entries with a minimal entry of three words. (The 2-word entry pointed to by the alternate CODES entry for one-to-N conversions is an exception.) The MULTIN entries associated with a starting internal character (index to CODES table) are linked in decreasing order of the number of chars. in the multiple string where the leading characters are the same.

The last MULTIN entry in a chain contains the conversion data for the single internal character. A MULTIN entry contains the second, third, etc. internal characters for the multiple conversion matching, packed two to a word beginning in the left half of the first word. The first full word following the characters is the match word for the string. This is any of the CODES table conversion entries except the multiple conversion. The next word following the match word is the no match word. This is either a link to another multiple conversion or the match word for the single starting internal character.

The NSPCE table is a continuous string of output characters packed two to a word. The characters are numbered 0,1,2 - beginning with the left character of the first word of the table.

The contents of the following Page 0 locations define the storage allocation for the conversion tables:

302 Starting address of CODES table

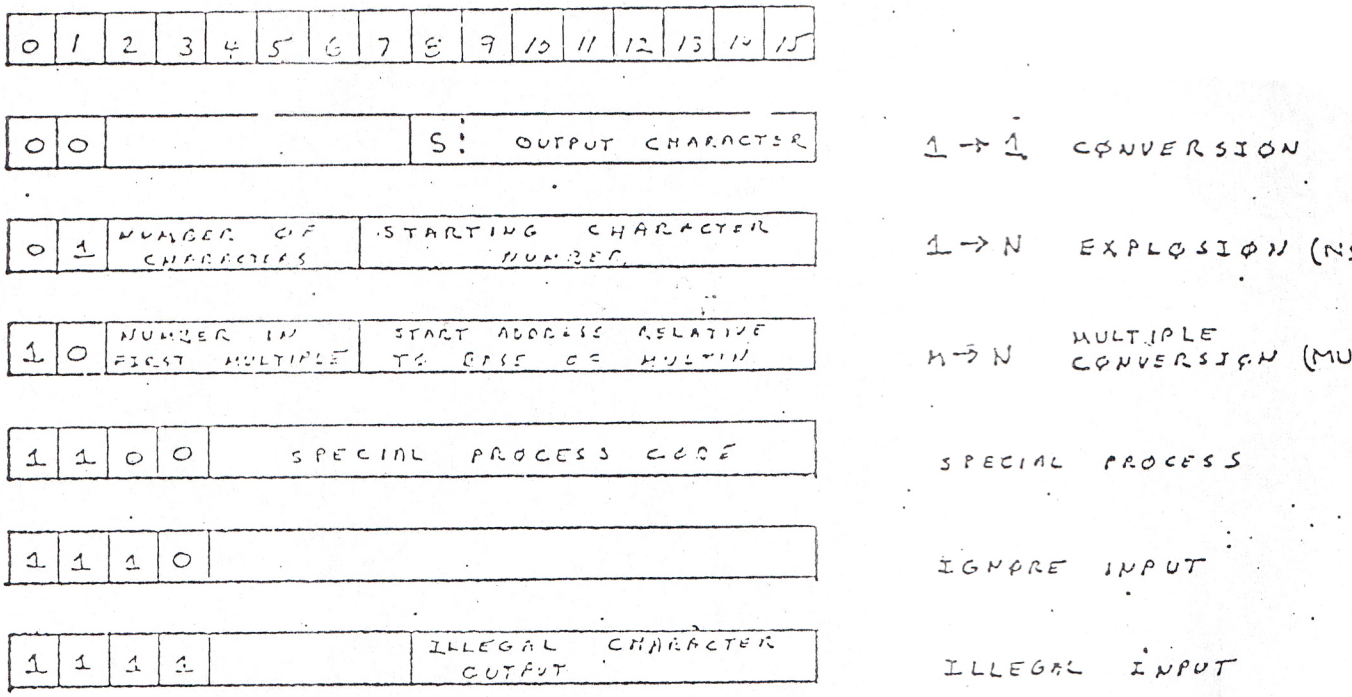
303 Starting address of MULTIN table

304 Starting address of NSPCE table

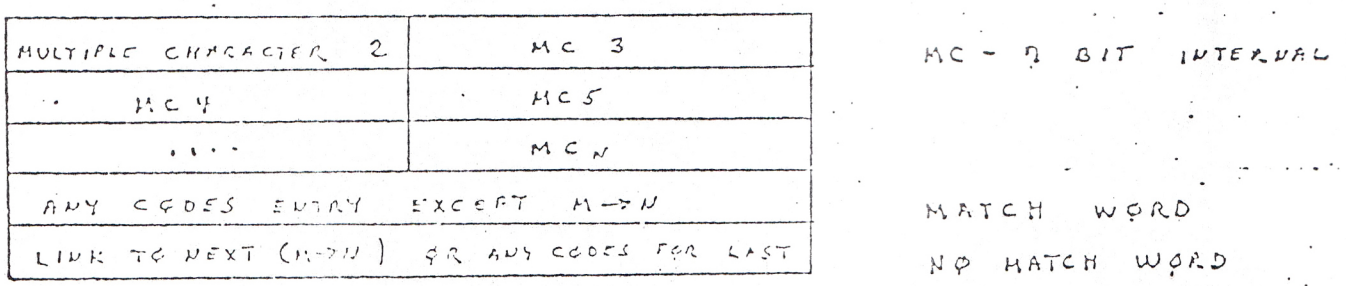
324 End address +1 of NSPCE Table (next available location).

325 Highest address the NSPCE Table can expand to.

CHARACTER TABLE - CODES

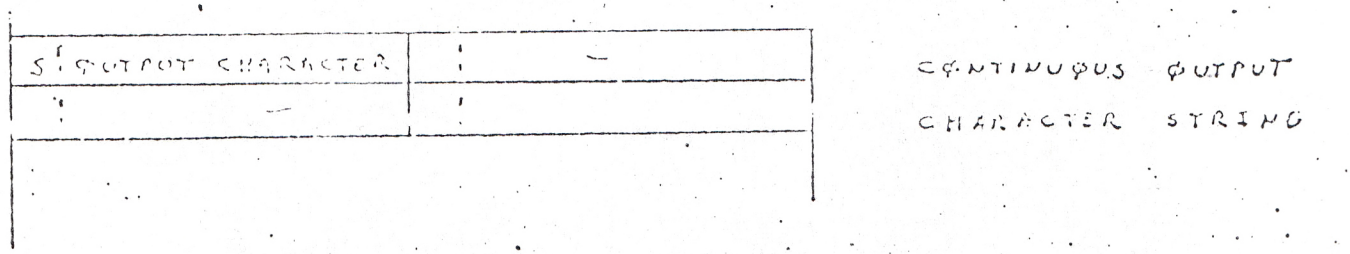

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 S: OUTPUT CHARACTER
1 → 1 CONVERSION
0 1 NUMBER OF CHARACTERS STARTING CHARACTER NUMBER
1 → N EXPLOSION (NS
1 0 NUMBER IN FIRST MULTIPLE START ADDRESS RELATIVE TO BASE OF MULTIN
M → N MULTIPLE CONVERSION (MUL
1 1 0 0 SPECIAL PROCESS CODE
SPECIAL PROCESS
1 1 1 0
IGNORE INPUT
1 1 1 1 ILLEGAL CHARACTER OUTPUT
ILLEGAL INPUT

MULTIPLE CONVERSION TABLE - MULTIN



MC - 7 BIT INTERNAL

EXPLOSION TABLE - NSPCE



S = SHIFT INDICATOR IF OUTPUT CODE < 8 BITS

S=1 UPPER CASE CHARACTER S=0 LOWER CASE CHARACTER